

**System and Method for Automatic
Population of Instant Messenger Lists**

BACKGROUND OF THE INVENTION

1. Technical Field

5 The present invention relates in general to a method and system for populating instant messenger (IM) lists. More particularly, the present invention relates to a system and method for populating IM lists based upon query results and user calendar data.

10 **2. Description of the Related Art**

Instant messaging has become a critical communications technology for many users and organizations. Instant messaging allows a user to create a "session" with one or more other users so that messages can be sent back and
15 forth contemporaneously. The flow of messages in an instant message session creates a dialog between the user and other users with whom the user has established a session. In this manner, quick messages can be transmitted without sending an email message or telephoning the other
20 user. Users can choose whether and when to respond to instant messages, much like an email message. However, unlike email messages, in instant messaging, a window is displayed on the user's display showing the messages between the user and another user.

25 A typical use of instant messaging is the ability to engage in "off-line" discussions during a telephone conference call. For example, in a negotiations conference call between two organizations, individuals within an

organization can use instant messaging to ask each other questions and make other comments about topics being discussed in the conference call without having to remember such questions and comments and discuss them after the call
5 has completed. Being able to have such "off-line" discussions enables the conference call participants to be better informed and more efficient so that more work is accomplished during the conference call and fewer "follow-up" calls are needed to resolve negotiation issues.

10 One challenge in using instant messaging, however, is the user's maintenance of his or her instant messaging contacts. These lists are often referred to as "buddy lists." While the user may have a permanent group of contacts in his buddy list, such as colleagues with whom he
15 frequently works, adding new contacts to the buddy list can be tedious and frustrating. Using the conference call negotiation example described above, the user would have to add any users with whom he wishes to have off-line discussions before he can establish an instant messaging
20 sessions with such users.

Another example is a consultant or other facilitator that may work with various groups of people in an organization at different times. If a consultant is working with a particular department for a period of time,
25 the users in that department would have to be added to the consultant's buddy list before the consultant could establish sessions with the users in that department. If the department being consulted by the consultant changes often, the consultant would spend considerable time adding
30 users to his buddy list. In addition, so that the

consultant's buddy list did not become unwieldy, the consultant would have to frequently remove previous clients and users from his buddy list.

What is needed, therefore, is a system and method that
5 automatically adds contacts to a user's buddy list based
upon data included in the user's calendar as well as
organizational data maintained in a database. What is
further needed, is a way of temporarily adding contacts to
the user's buddy list so that the temporarily added
10 contacts are automatically removed when no longer needed.

SUMMARY

It has been discovered that the aforementioned challenges are resolved using a system and method that automatically adds contacts to a user's instant messaging
5 buddy list. The contacts are added based upon existing database (i.e., directory) information as well as meeting information extracted from the user's calendar.

When a meeting is set up, a meeting notice is sent to all of the meeting attendees. These meetings are often
10 teleconference meetings with some or all of the attendees calling into the meeting from remote locations. The list of attendees is data that is included with the user's calendar. This list is used to automatically populate the user's buddy list. In one embodiment, a buddy list "group"
15 is initialized for the meeting. The user's buddy list includes a number of groups, for both meetings as well as other categories (e.g., "friends," "coworkers," etc.). In this manner, the user can expand and collapse various groups and select contacts included in the groups. For
20 example, the user can select a meeting group causing the list of attendees in the meeting to be displayed. The user can then select one or more of the displayed attendees in order to establish instant messaging sessions with the selected attendees.

25 Another way that contacts are added to a user's buddy list is by performing database queries. For example, a database directory may exist for an organization listing the various employees and other contacts for the organization along with information pertaining to the

employees and contacts, such as the department to which the employees and contacts belong. Queries are performed against the database directory returning groups of contacts, such as all the contacts within a particular
5 department. The group of contacts resulting from the query are then added to the user's buddy list.

Contacts added to the user's buddy list through the user's calendar (e.g., meeting attendees) or through database queries can be made permanent or temporary. When
10 contacts are added temporarily, the contacts are automatically removed after a specified period of time. For example, meeting attendees added to the user's buddy list might be automatically removed after the meeting has taken place.

15 The foregoing is a summary and thus contains, by necessity, simplifications, generalizations, and omissions of detail; consequently, those skilled in the art will appreciate that the summary is illustrative only and is not intended to be in any way limiting. Other aspects,
20 inventive features, and advantages of the present invention, as defined solely by the claims, will become apparent in the non-limiting detailed description set forth below.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may be better understood, and its numerous objects, features, and advantages made apparent to those skilled in the art by referencing the accompanying drawings. The use of the same reference
5 symbols in different drawings indicates similar or identical items.

Figure 1 is a network diagram showing the interaction between network components in automatically populating the user's buddy list;
10

Figure 2 is a screen diagram showing a user's buddy list with entries added from the user's calendar;

Figure 3 is a screen diagram showing a window with instant messenger contacts pulled from a database;

Figure 4 is a screen diagram showing the user selecting a calendar entry to add meeting participants to the user's buddy list;
15

Figure 5 is a screen diagram showing a window dialog used to add meeting attendees to the user's buddy list;

Figure 6 is a flowchart showing steps taken to add a group of contacts to a user's buddy list;
20

Figure 7 is a flowchart showing steps to create a buddy lists based upon attendees found in the user's calendar entries;

Figure 8 is a flowchart showing steps taken to remove temporary buddy groups and contacts from the user's buddy list; and
25

Figure 9 is a block diagram of an information handling system capable of implementing the present invention.

DETAILED DESCRIPTION

The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather,
5 any number of variations may fall within the scope of the invention which is defined in the claims following the description.

Figure 1 is a network diagram showing the interaction between network components in automatically populating the
10 user's buddy list. Client **100** is a user's computer system that access to calendar function **110**, instant messaging **120**, and computer network **150**, such as the Internet.

User's calendar **110** may be an application hosted by the user's computer system or may be a centralized
15 calendaring application hosted on a central computer system accessed through the computer network. The user's calendar includes data regarding participants to meetings and other events that the user is scheduled to attend, either telephonically or in person.

20 Instant messaging function **120** is an application that allows the user to send and receive instant messages with other users that are included in the user's buddy list **125**. The buddy list can include one or more buddy groups **130** for organizing the other users that the user can communicate
25 with using instant messaging function **120**. One way in which users and buddy groups are added to buddy list **125** is through extracting meeting participant information included in user's calendar function **110**. For example, the user can

decide to have participants for meetings added to the buddy list so that the user can establish an instant message session with other users attending the meeting. The users and buddy groups can be added to buddy list **125** on either a
5 permanent basis. Examples of other users include clients **155** and **160** that are computer systems for other users that are also connected to computer network **150**.

To establish an instant messaging session with another user, the user selects the other user from buddy list **125**.
10 The user can then instant message **140** to another user (e.g., client **155**, **160**, etc.). During the session, the user also receives instant messages **190** sent from one of the other users. The instant messages are displayed in a window of a display device included in client computer
15 system **100** and viewed by the user.

Another way that users are added to buddy list **125** is through use of a database or directory. As used herein, the term database includes both databases managed by a database management system (DBMS) as well as directories,
20 such as an LDAP directory. An example would be adding all members of a particular department to the user's buddy list. To add users from a database to the user's buddy list, the user sends query **145** through computer network **150** where it is received by server **175** as query request **170**.
25 Server **175** performs the requested query against database **180** and returns query results **185** to client **100** through computer network **150** which is received by client **100** as buddy list data **195** and added to buddy list **125**. While a separate server (server **175**) is shown accessing database
30 **180**, if client **100** has direct access to the database (e.g.,

database **180**), then the client can perform the query directly without needing the server to perform the request.

Figure 2 is a screen diagram showing a user's buddy list with entries added from the user's calendar. Window **200** shows the user's buddy list which has three different buddy groups: group **210** ("Team"), group **220** ("Friends/Colleagues"), and group **230** ("Calendar"). Groups **210** and **230** are shown as "expanded" (i.e., contacts within the groups viewable) while group **220** is shown as "collapsed" (i.e., contacts within the group not displayed until the group is selected).

Contacts can be added to the lists manually or automatically. For example, if contacts within the "Team" group are in the same department, the user can run a query to add some or all of the contacts (see **Figure 3** and corresponding text for an example and **Figure 6** and corresponding text for a flowchart of adding a list of contacts based on a query). The symbol next to the contact names indicates whether the contact is away, available, or unavailable. If the user wishes to begin an instant messaging session with one of the contacts, he simply selects the desired contact from the list.

Group **230** shows calendar entries, in particular, contacts for various meetings that are on the user's calendar. In the example shown, there are three meetings that appear in the calendar group: a planning meeting, a staff meeting, and a department meeting. Of these, the planning and department meeting are shown as collapsed, while the staff meeting is shown as expanded, showing the contacts. Similarly to the contacts within the "Team"

group, if the user wishes to begin an instant messaging session with a contact in the staff meeting, the user simply selects the desired contact. In the example shown, only one of the staff meeting contacts (Sue Abbot) is
5 currently available for an instant messaging session, the other are either away or unavailable. For further details on dialogs used to add calendar contacts to the user's instant messaging buddy list, see **Figures 4** and **5** and corresponding text. For details regarding the steps taken
10 to add contacts from calendar data, see the flowchart in **Figure 7** and corresponding text.

Figure 3 is a screen diagram showing a window with instant messenger contacts pulled from a database. Dialog window **300** includes text box **310** for entering a query,
15 combo box **320** for choosing a buddy group to which contacts are added, date picker **325** for choosing an optional expiration date for the group or contacts being added, and results window **330** for displaying the results of the user's query.

20 In the example shown, the user has selected a query of all records for a department named "XYZ" by entering a query of "Dept = 'XYZ'" in text box **310**. In another embodiment, the query is entered using a database wizard that asks the user a series of questions (i.e., what
25 field(s) to search, what criteria to search for, etc.). The user has selected that any contacts added as a result of the search will be added to the "Team" buddy group using combo box **320**. If the user selects combo box **320**, a list of the currently available buddy groups is presented to the
30 user in a drop down list and the user can select one of the

displayed buddy groups. The user can also type a new buddy group name in combo box **320** to request that contacts be added to a new buddy group. Date picker control **325** allows the user to select the drop down arrow whereupon a calendar is displayed allowing the user to select a date. Date picker control **325** also includes a checkbox. The expiration date is set if the user checks the checkbox, the expiration date is not set if the checkbox is left blank. Radio button control list **328** includes two radio buttons that allow the user to specify whether the expiration date (if provided) applies to the entire group or only to those contacts that are added to the group using the current dialog. In one embodiment, the management of the user's buddy groups is specified by a preset policy or configuration option.

When the user enters the query, records are retrieved from database **370**. In the example shown, records **380** are retrieved as these records are for individuals in the requested department ("XYZ"). The retrieved records are displayed in results window **330**. If the user wishes to add the displayed record to the selected buddy group, he selects "Add" command button **340** whereupon the displayed records are added to the buddy group (either temporarily or permanently as indicated by whether an expiration date has been set). On the other hand, if the user decides not to add the displayed contacts to the buddy list, the user selects "Cancel" command button **350** whereupon no additions are made to the user's buddy list and dialog window **300** closes.

Figure 4 is a screen diagram showing the user selecting a calendar entry to add meeting participants to the user's buddy list. Window **400** is a view of the user's calendar on which three entries appear for the given date. Entry **410** has been selected by the user (using a pointing device, keyboard, voice command, or any other selection means) and appears highlighted.

In response to selecting calendar entry **410**, item menu **420** appears with three options. Option **430** has been selected indicating that the user wishes to add participants (contacts) that are attending the selected meeting to his instant messaging buddy list. In response to selecting this option the contacts are added to the user's buddy list. In one embodiment, the contacts that are included in the calendar entry are automatically added to the buddy list or to a buddy list group (i.e., a new group could be created with a title indicative of the meeting). In another embodiment, a menu is presented to the user allowing the user to select options for how the contacts are to be added to his buddy list (for further details regarding this embodiment, see **Figure 5**).

Figure 5 is a screen diagram showing a window dialog used to add meeting attendees to the user's buddy list. Window **500** appears when the user requests adding contacts included in a calendar entry (e.g., an entry regarding a meeting with various attendees). Window **500** is provided to allow the user to determine where in the user's buddy list the contacts should be added, whether the additions are permanent or temporary, an expiration date for temporarily added contacts. Window **500** also allows the user to preview

the list of contacts before adding them to the user's buddy list.

Window **500** includes text box **510** which displays the calendar entry. In the example shown, the calendar entry
5 is for a department meeting being held on October 7, 2003. If this is not the entry for which the user wishes to add contacts, the user can exit by selecting "Cancel" command button **550**.

A default buddy group has been created with a name
10 that corresponds to the calendar entry. This buddy group name appears in combo box **520**. Any contacts included in the calendar entry will be added to this buddy group. However, by selecting combo box **520** a list of other buddy group names will appear and the user can select a different
15 buddy group where the contacts will be stored. The user can also type a new buddy group name in combo box **320** to request that contacts be added to a new buddy group. In one embodiment, the management of the user's buddy groups is specified by a preset policy or configuration option.

20 Date picker control **525** allows the user to select the drop down arrow whereupon a calendar is displayed allowing the user to select a date. Date picker control **525** also includes a checkbox. The expiration date is set if the user checks the checkbox, the expiration date is not set if
25 the checkbox is left blank. By default, the expiration date is set for the day following the calendar entry and the contacts will be automatically removed from the user's buddy list upon the expiration date.

Radio button control list **528** includes two radio buttons that allow the user to specify whether the expiration date applies to the entire group or only to those contacts that are added to the group using the
5 current dialog. By default, the option has been selected that the entire group will be removed as the default group name created by the system corresponds to the calendar entry.

Contacts included in the selected calendar entry are
10 retrieved from calendar database **570**. Calendar database includes many fields about each of the calendar entries, including attendees list **580**. Attendees list **580** is retrieved and displayed to the user in window **530**. The user can review the contacts that appear in window **530** and
15 decide whether to add them to his buddy list, which buddy group to add them to, and whether the additions are permanent or temporary. When the user has made his decisions and selections, he selects "Add" command button **540** to add the contacts to the selected buddy group. If
20 the user decides not to add the contacts, he selects "Cancel" command button whereupon the contacts are not added to the user's buddy list and window **500** closes.

Figure 6 is a flowchart showing steps taken to add a group of contacts to a user's buddy list. Processing
25 commences at **600** whereupon, at step **605**, the name of the buddy group, the requested query, and (optionally) an expiration date are provided by user **610**. In one embodiment, a database wizard is used by the user to formulate the user's query so that the user does not need

to have detailed knowledge of the field names or query syntax.

At step **615**, the query is sent to server **620** for processing. Server **620** processes the requested query using database **625** and returns the results of the query where they are received at step **630**. Steps **615** through **630** assume that the user either does not have direct access to the database or that the database is remote from the user's computer system. If this is not the case, then the user's computer system can directly perform the query using a directly accessible database.

A determination is made as to whether the results returned from the query are empty indicating that no records (contacts) satisfied the user's query request (decision **635**). If the list is empty, decision **635** branches to "yes" branch **640** whereupon, at step **645**, an error message is returned to the user indicating that no records matched the user's query request and processing ends at **650**. On the other hand, if the list was not empty, decision **635** branches to "no" branch **655** in order to process the list.

A determination is made as to whether the buddy group name to which the contacts are being added is a new, or existing, buddy group (decision **660**). If it is a new buddy group, decision **660** branches to "yes" branch **665** whereupon, at step **670**, the new buddy group is created in buddy list data store **675**. In addition, if the new buddy group is temporary, the expiration date of the new buddy group is written to the new buddy group data. At step **680**, the

entries received as a result of the query are added to the newly created buddy group stored in data store **675**.

Returning to decision **660**, if the entries are being added to an existing buddy group, decision **660** branches to
5 "no" branch **685** whereupon, at step **690**, the entries received as a result of the query are added to the existing buddy group stored in data store **675**. In addition, if an expiration date has been specified for one or more of the contacts being added, the expiration date is written to
10 buddy list data store **675** and associated with the temporary contacts. Processing thereafter ends at **695**.

Figure 7 is a flowchart showing steps to create a buddy lists based upon attendees found in the user's calendar entries. Processing commences at **700** whereupon, at step
15 **705**, one or more calendar entries are selected from calendar data store **710** (calendar data store **710** including participant contact data corresponding to various calendar entries).

At step **715**, the first selected calendar entry is
20 selected. At step **720**, a buddy group name is either created if the buddy group is new (i.e., the name corresponding to the calendar entry), or an existing buddy group within buddy list **675** is located. In one embodiment, the management of the user's buddy groups is specified by a
25 preset policy or configuration option.

If the buddy group is a temporary buddy group, the expiration date corresponding to the buddy group is written to buddy list **675** at step **730**. The first contact from the selected calendar entry is selected at step **735**. The

selected contact is added to the selected buddy group at step **740**. If the contact is being added temporarily, the expiration date that applies to the contact is also recorded along with the contact information.

5 A determination is made as to whether there are more contacts in the calendar entry (e.g., meeting, etc.) to add to the user's buddy list (decision **750**). If there are more contacts to add, decision **750** branches to "yes" branch **755** whereupon, at step **760**, the next contact in the calendar
10 entry is retrieved from calendar data store **710** and processing loops back to add the contact to the user's buddy list. This looping continues until all contacts in the calendar entry have been processed, at which point decision **750** branches to "no" branch **765**.

15 A determination is made as to whether there are more calendar entries, such as meetings, that have been selected to process (decision **770**). If there are more selected calendar entries to process, decision **770** branches to "yes" branch **775** whereupon, at step **780**, the next calendar entry
20 is selected from calendar data store **710** and processing loops back to process the selected calendar entry. This looping continues until all selected calendar entries have been processed, at which point decision **770** branches to "no" branch **785** and processing ends at **795**.

25 **Figure 8** is a flowchart showing steps taken to remove temporary buddy groups and contacts from the user's buddy list. As explained in **Figures 7** and **8** above, contacts and buddy groups can be stored temporarily. For example, contacts for a given meeting can be stored in the user's
30 buddy list until the date of the meeting is past.

Processing commences at **800** whereupon, at step **810** the first buddy group within buddy list **675** is selected. A determination is made as to whether the entire buddy group is a temporary buddy group (decision **820**). If the selected
5 buddy group is temporary (e.g., a group containing the contacts for a particular meeting, etc.), decision **820** branches to "yes" branch **825** whereupon another determination is made as to whether the current date (i.e., system date) is greater than (after) the buddy group
10 expiration date (decision **830**). If the current date is greater than the buddy group expiration date, decision **830** branches to "yes" branch **835** whereupon, at step **840**, the buddy group is removed from buddy list **675**. On the other hand, if the current date is not greater than the buddy
15 list expiration date (i.e., the buddy group has not yet expired), decision **830** branches to "no" branch **855** bypassing step **840**.

Returning to decision **820**, if the selected buddy group is not a temporary buddy group, decision **820** branches to
20 "no" branch **845** whereupon, at step **850**, an individual expired contacts within the group are removed from the buddy group. For example, a temporary worker or visitor may be added to one of the user's permanent buddy groups for the duration of their work. When the date assigned to
25 the temporary worker/visitor has been reached, the contact information is removed, at step **850**, from one of the user's permanent buddy groups.

A determination is made as to whether there are more buddy groups to process (decision **870**). If there are more
30 buddy groups to process, decision **870** branches to "yes"

branch **875** whereupon, at step **880**, the next buddy group within the user's buddy list is selected from buddy list data store **675** and processing loops back to determine whether the buddy group, or temporary contacts within the
5 buddy group, should be removed. This looping continues until all buddy groups have been processed, at which point decision **870** branches to "no" branch **885** and processing ends at **895**.

Figure 9 illustrates information handling system **901**
10 which is a simplified example of a computer system capable of performing the computing operations described herein. Computer system **901** includes processor **900** which is coupled to host bus **902**. A level two (L2) cache memory **904** is also coupled to host bus **902**. Host-to-PCI bridge **906** is coupled
15 to main memory **908**, includes cache memory and main memory control functions, and provides bus control to handle transfers among PCI bus **910**, processor **900**, L2 cache **904**, main memory **908**, and host bus **902**. Main memory **908** is coupled to Host-to-PCI bridge **906** as well as host bus **902**.
20 Devices used solely by host processor(s) **900**, such as LAN card **930**, are coupled to PCI bus **910**. Service Processor Interface and ISA Access Pass-through **912** provides an interface between PCI bus **910** and PCI bus **914**. In this manner, PCI bus **914** is insulated from PCI bus **910**.
25 Devices, such as flash memory **918**, are coupled to PCI bus **914**. In one implementation, flash memory **918** includes BIOS code that incorporates the necessary processor executable code for a variety of low-level system functions and system boot functions.

PCI bus **914** provides an interface for a variety of devices that are shared by host processor(s) **900** and Service Processor **916** including, for example, flash memory **918**. PCI-to-ISA bridge **935** provides bus control to handle
5 transfers between PCI bus **914** and ISA bus **940**, universal serial bus (USB) functionality **945**, power management functionality **955**, and can include other functional elements not shown, such as a real-time clock (RTC), DMA control, interrupt support, and system management bus
10 support. Nonvolatile RAM **920** is attached to ISA Bus **940**. Service Processor **916** includes JTAG and I2C busses **922** for communication with processor(s) **900** during initialization steps. JTAG/I2C busses **922** are also coupled to L2 cache **904**, Host-to-PCI bridge **906**, and main memory **908** providing
15 a communications path between the processor, the Service Processor, the L2 cache, the Host-to-PCI bridge, and the main memory. Service Processor **916** also has access to system power resources for powering down information handling device **901**.

20 Peripheral devices and input/output (I/O) devices can be attached to various interfaces (e.g., parallel interface **962**, serial interface **964**, keyboard interface **968**, and mouse interface **970** coupled to ISA bus **940**. Alternatively, many I/O devices can be accommodated by a super I/O
25 controller (not shown) attached to ISA bus **940**.

In order to attach computer system **901** to another computer system to copy files over a network, LAN card **930** is coupled to PCI bus **910**. Similarly, to connect computer system **901** to an ISP to connect to the Internet using a

telephone line connection, modem **975** is connected to serial port **964** and PCI-to-ISA Bridge **935**.

While the computer system described in **Figure 9** is capable of executing the processes described herein, this
5 computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the processes described herein.

While the computer system described in **Figure 9** is
10 capable of executing the invention described herein, this computer system is simply one example of a computer system. Those skilled in the art will appreciate that many other computer system designs are capable of performing the invention described herein.

15 One of the preferred implementations of the invention is an application, namely, a set of instructions (program code) in a code module which may, for example, be resident in the random access memory of the computer. Until required by the computer, the set of instructions may be
20 stored in another computer memory, for example, on a hard disk drive, or in removable storage such as an optical disk (for eventual use in a CD ROM) or floppy disk (for eventual use in a floppy disk drive), or downloaded via the Internet or other computer network. Thus, the present invention may
25 be implemented as a computer program product for use in a computer. In addition, although the various methods described are conveniently implemented in a general purpose computer selectively activated or reconfigured by software, one of ordinary skill in the art would also recognize that
30 such methods may be carried out in hardware, in firmware,

or in more specialized apparatus constructed to perform the required method steps.

While particular embodiments of the present invention have been shown and described, it will be obvious to those skilled in the art that, based upon the teachings herein, changes and modifications may be made without departing from this invention and its broader aspects and, therefore, the appended claims are to encompass within their scope all such changes and modifications as are within the true spirit and scope of this invention. Furthermore, it is to be understood that the invention is solely defined by the appended claims. It will be understood by those with skill in the art that if a specific number of an introduced claim element is intended, such intent will be explicitly recited in the claim, and in the absence of such recitation no such limitation is present. For a non-limiting example, as an aid to understanding, the following appended claims contain usage of the introductory phrases "at least one" and "one or more" to introduce claim elements. However, the use of such phrases should not be construed to imply that the introduction of a claim element by the indefinite articles "a" or "an" limits any particular claim containing such introduced claim element to inventions containing only one such element, even when the same claim includes the introductory phrases "one or more" or "at least one" and indefinite articles such as "a" or "an"; the same holds true for the use in the claims of definite articles.